04-19-06

AF
?W

Atty. Docket Nbr. RSW920010125US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of Robert H. High, Jr., et al.

Serial Nbr:	09/943,618	Filed:	August 30, 2001
For:	Role-Permission Model for Security Policy Administration and Enforcement		
Art Unit:	2131	Examiner:	Arezoo Sherkat

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF IN SUPPORT OF APPEAL FROM THE PRIMARY EXAMINER TO THE BOARD OF APPEALS

Sir:

Appellants herewith submit an Appeal Brief in support of the appeal to the Board of Appeals from the decision dated November 22, 2005 of the Primary Examiner finally rejecting all Claims 3 - 12 and 19 - 26.

The appeal brief fee of $500.00 is:

□ Enclosed.
□ Not required. (Fee paid in prior appeal.)
☒ Charged to Deposit Account No. **09/0461**. A duplicate copy of this sheet is enclosed.

Oral Hearing is:

☒ Not requested.
□ Requested. See first paragraph of accompanying appeal brief.

Date: April 18, 2006

Respectfully submitted,

By Marcia L. Doubet
Marcia L. Doubet, Attorney for Appellants
Registration No. 40,999

Customer Number 43168
Telephone: 407-343-7586; Fax: 407-343-7587

Serial No. 09/943,618

Atty. Docket No. RSW920010125US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of Robert H. High, Jr., et al.

Serial Nbr: 09/943,618

Filed: August 30, 2001

For: Role-Permission Model for Security Policy Administration and Enforcement

Art Unit: 2131

Examiner: Arezoo Sherkat

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

EXPRESS MAIL CERTIFICATE

"Express Mail" label number: **EQ049387228US** Date of Deposit: April 18, 2006

WE REQUEST THE DATE OF DEPOSIT AS THE DATE FILED.

I hereby certify that the following enclosed or attached papers and fee

(1) Transmittal of Patent Appeal Brief, requesting payment of Appeal Brief Fee from deposit account (1 page, in duplicate)
(2) Appeal Brief (28 pages)
(3) Business Reply Postcard

are being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. §1.10 on the date indicated above and are addressed to Mail Stop Appeal Brief - Patent, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450.

__Marcia L. Doubet______________
(Name of person mailing paper or fee)

Marcia L. Doubet
(Signature of person mailing paper or fee)

Attorney Docket RSW920010125US1

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

In re application of Robert H. High, Jr., et al. April 18, 2006

Serial Nbr: 09/943,618

Filed: August 30, 2001

For: Role-Permission Model for Security Policy Administration and Enforcement

Art Unit: 2131

Examiner: Arezoo Sherkat

APPELLANTS' BRIEF ON APPEAL

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

Sir:

This is an Appeal seeking reversal of the decision of the Primary Examiner, finally rejecting all current claims of the subject patent application.

1) REAL PARTY IN INTEREST

The real party in interest is the Assignee, International Business Machines Corporation ("IBM").

2) RELATED APPEALS AND INTERFERENCES

Appellants, the Appellants' legal representative, and the assignee, have no personal knowledge of any other appeals or interferences which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

3) STATUS OF CLAIMS

Claims 3 - 12 and 19 - 26 stand rejected. Claims 1 - 2 and 13 - 18 have been cancelled from the application without prejudice. Claims 3 - 12 and 19 - 26 are under appeal.

4) STATUS OF AMENDMENTS

No Amendments were filed after receiving the Final Rejection mailed on November 22, 2005.

5) SUMMARY OF CLAIMED SUBJECT MATTER

1. Appellants' independent Claim 19 specifies elements of "storing, in a security repository, a plurality of <u>security objects</u>, wherein each of the security objects <u>corresponds to a single role</u>" (Claim 19, lines 3 - 4, emphasis added); "specifying, in each of the security objects, <u>all permissions granted to the corresponding role</u>, wherein each of the specified permissions identifies at least one resource and, for each resource, at least one action that can be performed on the resource by subjects granted the corresponding role, wherein selected ones of the

resources are identified in the specified permissions of more than one of the security objects and wherein the specified permissions for at least one of the security objects identifies a plurality of resources and for each of the plurality of resources, at least one of the actions" (Claim 19, lines 5 - 11, emphasis added); and "using the stored security objects to determine whether run-time requests for performing actions on the resources can be granted" (Claim 19, lines 12 - 13). Independent Claims 23 and 25 specify similar limitations.

2. In other words, each security object corresponds to a role (Specification, p. 17, lines 14 and 17). For example, with reference to Claim 19, lines 3 - 4, a security object may be stored for a "Teller" role (**Fig. 5**, first row; **Fig. 4**, see arrow from "Role" to "SecurityObject" **420**). Within this "Teller" role security object, "all permissions granted" to users associated with the "Teller" role are specified (Claim 19, lines 5 - 6). So, for example, if Tellers can invoke a "getBalance" method on an "Account" resource and a "deposit" method on this "Account" resource, as well as a ".GET" request on a resource identified as "/finance/account", then those granted permissions are all specified in the security object for the Teller role (see **Fig. 5**, intersection of first row with first, second, and fifth columns; Specification, p. 19, lines 1 - 2, lines 7 - 8, and lines 13 - 15). Each of the permissions specified for each role "identifies at least one resource" (Claim 19, line 6) "and, for each resource, at least one action that can be performed on the resource by subjects granted the corresponding role" (Claim 19, lines 6 - 8). In the Teller example, the "identified at least one resource" from Claim 19, line 6 comprises (1) the "Account" resource and (2) the "/finance/account" resource, and the "at least one action ..." for each of these resources (from Claim 19, lines 6 - 8) comprises the (i) "getBalance" and (ii) "deposit" methods, for the

"Account" resource, and the ".GET" request for the "/finance/account" resource. The "selected ones of the resources are identified ..." language in Claim 19, lines 8 - 9 indicates that a resource may have permissions specified in more than one of the security objects. (That is, more than one Role can be given permission to access a particular resource.) For example, the "Account" resource is "identified in the specified permissions" of the "Teller" role and also in the permissions of the "Cashier" and "Super" (i.e., Supervisor) roles, as shown by the first through third rows, and first through third columns, in **Fig. 5**. The "specified permissions for at least one of the security objects identifies a plurality of resources" language in Claim 19, lines 9 - 10 indicates that at least one of the roles has permission to access more than one resource. For example, the permissions specified in the Teller-role security object identify permissions for (1) the "Account" resource and (2) the "/finance/account" resource, as discussed above. The "for each of the plurality of resources, at least one of the actions" language in Claim 19, lines 10 - 11 indicates that, for example, the Teller-role security object specifies actions of (i) "getBalance" and (ii) "deposit" on the "Account" resource, as also discussed above.

3. Dependent Claim 3 specifies "wherein at least one of the resources is an executable method". For example, the "getBalance" method on the "Account" Enterprise JavaBean ("EJB") is treated as a resource in this instance. Dependent Claim 4 specifies "wherein at least one of the resources is a column of a database table" and dependent Claim 5 specifies "wherein at least one of the resources is a row of a database table". For these instances, the role specifies what column(s) or row(s), respectively, of a database table can be accessed by subjects having that particular role (Specification, p. 24, lines 5 - 8).

4. Dependent Claim 6 specifies "wherein at least one of the resources is a file and each of the at least one actions identified for the at least one resource are file access operations that can be performed on the file". For example, permission to delete a selected file on a remote server may be granted to subjects having a specific role, where the resource is the file and the action is "delete" (Specification, p. 24, lines 10 - 11). Dependent Claim 7 specifies "wherein at least one of the resources is a function call to a function of an executable program". Specification, p. 24, lines 9 - 10. This "function call" language is similar to the "getBalance" method invocation in an object-oriented environment. Dependent Claim 8 specifies "wherein at least one of the resources is an Enterprise JavaBean ("EJB") and each of the at least one actions identified for the at least one resource are methods that can be performed on the EJB". In this instance, the "Account" EJB is an example of the resource, and the "actions"/"methods" are exemplified by "getBalance", "deposit", and "closeAccount", as illustrated in the first three columns of table **500** in **Fig. 5**.

5. Dependent Claim 9 specifies "wherein at least one of the resources is a servlet and each of the at least one actions identified for the at least one resource are methods that can be performed by the servlet". Dependent Claim 10 specifies "wherein at least one of the resources is a Uniform Resource Identifier ("URI") and each of the at least one actions identified for the at least one resource are methods which reference the URI", and dependent Claim 11 specifies "wherein at least one of the resources is a JavaServer Page ("JSP") and each of the at least one actions identified for the at least one resource are methods referenced from the JSP". Dependent Claim 12 specifies "wherein at least one of the resources is any resource that is expressible to the

security system and each of the at least one actions identified for the at least one resource are selected from a set of actions that are permitted on that resource". These are all various types of resource/action combinations that may be protected using the present invention. Specification, p. 24, lines 11 - 12.

6. Dependent Claim 20 specifies further details of the "using the stored security objects to determine whether run-time requests for performing actions on the resources can be granted" element of independent Claim 19. In particular, these further details comprise "determining, for the run-time request, a requester from which the request was received, and a particular action being requested on a particular resource" (Claim 20, lines 3 - 4, emphasis added); "determining one or more roles granted to the requester" (Claim 20, line 5, emphasis added); and "until determining that the request can be granted or exhausting the determined roles, iteratively accessing the security object corresponding to each one of the determined roles and if the accessed security object identifies the requested action on the requested resource, then determining that the request can be granted" (Claim 20, lines 6 - 9, emphasis added). See also Specification, p. 22, line 8 - p. 23, line 1, where these steps are discussed. So, for example, if a user "Bob" issues an invocation request for a "deposit" method/action on an "Account" resource, then the roles granted to Bob (the requester) are determined, and each of these granted roles is iteratively checked until finding a role for which the specified permissions include this particular "deposit" action on this particular "Account" resource; if such a role is found for Bob, then his invocation request is granted. (Specification, p. 19, lines 13 - 15.) Dependent Claim 24 specifies analogous limitations.

7. Dependent Claim 21 specifies further details for the "determining one or more roles" element of Claim 20. These further details comprise "using an identification of the requester as a user identification to consult a mapping that specifies, for each of a plurality of subjects, one or more roles associated therewith, wherein each of the subjects is specified as at least one of (1) an identification of one or more users and (2) an identification of one or more user groups, thereby determining each role associated with the identification of the requester" (Claim 21, lines 3 - 7); "determining one or more user groups of which the requester is a member" (Claim 21, line 8); and "using each of the determined user groups as a user group identification to consult the mapping, thereby determining each role associated with the determined user groups" (Claim 21, lines 9 - 10). Accordingly, the user is identified, as are any user groups of which the user is a member, and the roles granted to this user are the union of the user's roles and the roles of the user groups of which this user is a member.

8. Dependent Claim 22 specifies further details of the "using the stored security objects ..." element of independent Claim 19, and in particular, "determining, for the run-time request, a requester from which the request was received, and a particular action being requested on a particular resource" (Claim 22, lines 3 - 4); and "determining that the run-time request can be granted only if the requester has been granted at least one of the roles which is required, according to the stored security objects, to perform the requested action on the requested resource" (Claim 22, lines 5 - 7, emphasis added). Dependent Claim 26 specifies analogous limitations.

9. Independent Claim 23 and dependent Claim 24 include means plus function terminology. Structure, material, or acts supporting this terminology are described in Appellants' specification, as will now be described.

10. With regard to the "means for specifying" element of independent Claim 23, the text on p. 16, lines 11 - 17 describes specifying role-based information for security objects; see also **Block 420** of **Fig. 4** and its corresponding text on p. 17, lines 11 - 16. Page 17, lines 5 - 10 discusses a scenario where a particular role includes information for more than one resource. For the "means for using" element of independent Claim 23, see the text on p. 22, line 8 - p. 23, line 1, where run-time analysis of requests is discussed.

6) GROUND OF REJECTION TO BE REVIEWED ON APPEAL

11. The ground of rejection presented for review is whether Claims 3 - 12 and 19 - 26 are anticipated under 35 U.S.C. §102(b) by Barkley et al. (U.S. 6,202,066).

7) ARGUMENT

12. Page 2 of the Office Action dated November 22, 2005 (hereinafter, "the Office Action") states that Claims 3 - 12 and 19 - 26 are rejected under 35 U.S.C. §102(b) as being anticipated by U. S. Patent 6,202,066 to Barkley et al. (hereinafter, "Barkley"). Of these claims, the independent claims are 19, 23, and 25.

13. Appellants respectfully submit that a *prima facie* case of anticipation under 35 U.S.C.

§102 has not been made out as to their Claims 3 - 12 and 19 - 26. Section 706.02 of the MPEP, "Rejection on Prior Art", states in Section IV, "Distinction Between 35 U.S.C. 102 and 103", the requirements for establishing a *prima facie* case of anticipation under this statute, noting that "... for anticipation under 35 U.S.C. 102, the reference must teach <u>every aspect</u> of the claimed invention either explicitly or impliedly" (emphasis added). This requirement is also stated in MPEP §2131, "Anticipation -- Application of 35 U.S.C. 102(a), (b), and (e)", which states (in its final paragraph) "A claim is anticipated only if <u>each and every element</u> as set forth in the claim is found, either expressly or inherently described, in a single prior art reference", quoting *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987), emphasis added. This final paragraph of MPEP §2131 also states "The elements <u>must be arranged</u> as required by the claim ...", quoting *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990), emphasis added.

14. Furthermore, Appellants are entitled to have <u>all words</u> of their claimed invention considered when determining patentability. See Section 2143.03 of the MPEP, "All Claim Limitations Must Be Taught or Suggested", referencing ***In re Wilson***, 165 USPQ 494, 496 (C.C.P.A. 1970), which stated "*All words* in a claim must be considered in judging the patentability of that claim against the prior art." (emphasis added).

15. The burden for rebutting a rejection under 35 U.S.C. 102 does not pass to Appellants until a *prima facie* case of anticipation has been made out. See *In re Bass*, 177 USPQ 178, 186 (C.C.P.A. 1973), which held:

> From the evidence available to it, the initial burden of making out a prima facie case of prior invention is on the Patent Office. . . . When the Patent Office has made out a prima facie case of priority the burden would then shift to the applicant to rebut it.

Accordingly, Appellants respectfully submit that the burden has not passed. For the sake of expediency, Appellants will, however, provide a rebuttal herein of the analysis provided in the Office Action.

7.1) Rejection of Independent Claims 19, 23, and 25

16. Appellants respectfully submit that Barkley <u>fails to teach</u> all limitations of their independent Claims 19, 23, and 25 -- and in particular, does <u>not</u> teach "each and every element" or "all words" of these claims. The Office Action analysis therefore fails to make out a *prima facie* case of anticipation, in violation of the above-quoted MPEP §706.02, §2131, and §2143.03, as will now be demonstrated.

17. Pages 2 (final 4 lines) - 3 (entire page) of the Office Action analyze Appellants' independent Claims 19, 23, and 25. This analysis will now be described.

18. The Office Action cites Barkley's Object Access Types ("OATs") as being analogous to Appellants' security objects. See, for example, Office Action p. 3, lines 15 - 20. Appellants respectfully submit that these OATs cannot properly be equated to their claim language, as Barkley's OATs are <u>different from</u> Appellants' claim limitations.

19. With regard to the first element of Appellants' Claim 19, Barkley does not teach "wherein each of the security objects [stored in a security repository] corresponds to a single role" (Claim 19, lines 3 - 4, emphasis added). For example, one of Appellants' security objects corresponds to the "Teller" role, which a different security object corresponds to the "Cashier" role and yet another security object corresponds to the "Super" (Supervisor) role. Specification, p. 17, lines 4 - 6; p. 18, lines 9 - 15; p. 19, line 18; **Fig. 5**.

20. Instead, Barkley teaches that his OATs may specify information for more than one role. See, for example, the following citations:

- ♦ col. 5, lines 5 - 14, describing a scenario where "members of a first role are given a first level of access or permissions to a first set of files or objects, while those designated to a second role are granted a different second level of access to the same set of files. An OAT [note, singular: one OAT] is then created associating the first role ... and the second role ... [that is, the same OAT is used for both roles] The OAT [again, note singular: one OAT] is then assigned ..." (emphasis added);
- ♦ col. 7, lines 4 - 5, referring to a "list of roles [note, plural] or groups" (emphasis added) that can be associated with an object via a 3-tuple used for specifying each of Barkley's OATs (where the 3-tuple is further discussed in col. 6, line 61 - col. 7, line 3);
- ♦ col. 9, lines 2 - 3, stating that a role can be removed from an OAT (which, by implication, indicates that the OAT originally contained more than one role);

- col. 12, lines 42 - 45, "... all of the objects assigned to a given OAT may be accessed identically by members of each of the roles [note, plural use of "roles"] assigned to that [single, "given"] OAT ..." (emphasis added);
- Table I in col. 11, lines 10 - 19, where each of the columns of Table I represents a single OAT (such as "accounts") and each of the rows identifies one of a plurality of roles that may be represented in each of the OATs. The OAT for "accounts", for example, in the first column represents 4 of the 5 roles from Table I (namely, the "account_rep", "branch_manager", "financial_advisor", and "teller" roles which each have some permission-related notation in this first column). This is described in the corresponding text at col. 11, line 63 - col. 12, line 32. In this text, the "account_rep" role's permissions for the "account" OAT are discussed at col. 11, line 63 - col. 12, line 6; the permissions for the "branch_manager" role of the "account" OAT are discussed at col. 12, lines 7 - 9; the "financial_advisor" role's permissions for the "account" OAT are discussed at col. 12, lines 13 - 26; and the permissions for the "teller" role of the "account" OAT are discussed at col. 12, lines 27 - 32;
- col. 12, lines 44 - 45 refer to "each of the roles assigned to that OAT", using "roles" in the plural and "OAT" in the singular; and
- Claim 1 in col. 14, lines 35 - 37, "each OAT being a separate entity for associating one or more objects with one or more roles ...", emphasis added.

21. Specifying one OAT that addresses multiple roles, as taught by Barkley and evidenced by

(at least) the above-cited textual references in paragraph 20, is distinct from Appellants' security objects which each correspond to a single role (i.e., "wherein each of the security objects corresponds to a single role", Claim 19, lines 3 - 4).

22. Accordingly, as demonstrated by paragraphs 19 - 21 herein, Barkley fails to teach the first element of Appellants' independent Claim 19. Independent Claims 23 and 25 specify analogous limitations.

23. With regard to the second element of Appellants' Claim 19, Barkley does not teach "specifying, in each of the security objects, all permissions granted to the corresponding role" (Claim 19, lines 5 - 6, emphasis added). For example, all permissions granted to the Teller role are specified in the security object that corresponds to this Teller role. Specification, p. 17, lines 5 - 10.

24. Instead, Barkley teaches that permissions for a particular role can be split across multiple OATs. See, for example, the following citations:

- ♦ col. 7, lines 53 - 55, stating "changes in the permissions granted to a particular role can be implemented simply by changing ... the corresponding OATs" (emphasis added; note, plural use of OATs when referring to a single, "particular", role); and
- ♦ col. 12, lines 46 - 49, stating "... the members of a given role may be assigned differing permissions ... by being assigned membership in differing OATs"

(emphasis added; a single, "given" role is addressed in multiple, "differing", OATs).

25. Specifying the permissions for a particular role in more than one OAT, as taught by Barkley and evidenced by (at least) the above-cited textual references in paragraph 24, is distinct from specifying "all permissions granted to the corresponding role" in one (i.e., "each") of the security objects, as specified in Appellants' claim language ("specifying, in each of the security objects, all permissions granted to the [note, singular "the"] corresponding role"; Claim 19, lines 5 - 6, emphasis added).

26. Furthermore, still referring to the second element of Appellants' Claim 19, Barkley does not teach the limitations on lines 9 - 11, namely "wherein the specified permissions for at least one of the security objects [i.e., for at least one of the roles, since each security object corresponds to a single role, as specified in Claim 19, lines 3 - 4] identifies a plurality of resources and for each of the plurality of resources, at least one of the actions" (Claim 19, lines 9 - 11, emphasis added). For example, with reference to Appellants' sample "Teller" role presented in the first row of table **500** in **Fig. 5**, this sample role is illustrated as having permissions for a plurality of resources, namely for (1) the "Accounts" resource (first and second columns) and also for (2) the "/finance/account" resource (fifth column).

27. Instead, Barkley teaches use of 3-tuples that, for a particular role, specify a single object or resource (and a plurality of actions thereupon). See, for example, the following citations:

- col. 6, lines 61 - 62, stating “This association can be represented as a 3-tuple: (role or group; object [note, singular, “object” within a role]; {permitted operations on object [note, singular use of “object”]}” -- and note also that the “curly brackets” notation is used to specify multiple permitted actions within the tuple but, notably, this curly brackets notation is not used for the “object” specification in the middle of the tuple, thus indicating that multiple objects were not intended for a particular role; and
- col. 6, lines 63 - 65, stating “... a user assigned to role ... is authorized to perform operation on object [note, singular, “object”] ...”.

28. As mentioned above in paragraph 27, it is noted that Barkley has explicitly used the “curly brackets” notation in the syntax specification provided at col. 6, lines 61 - 62 to indicate that each 3-tuple may specify multiple permitted operations on an object. Accordingly, failure to use this same syntax for the *object* is consistent with the description thereof in col. 6, lines 61 - 64, where “object” is discussed in the singular.

29. Barkley presents an alternative, “isomorphic”, representation of his 3-tuple at col. 6, line 66 - col. 7, line 3, where the “role/group” and “object” elements are switched in position within the tuple. This alternative representation thus specifies an organization based on objects, which is distinct from Appellants’ role-based organization. (Note also that Appellants have discussed drawbacks of prior art techniques whereby permissions are organized according to protected resources in their Specification on p. 3, lines 14 - 16; p. 4, lines 8 - 10; and p. 14, lines 5 - 14.

See also p. 5, lines 16 - 17, stating that Appellants' invention enforces security policy "for each role instead of for each protected resource", emphasis added. The text on p. 18, lines 5 - 15 also contrasts the prior art resource-based approach to Appellants' role-based approach. Namely, the prior art resource-based organization -- which may equivalently be viewed as an organization according to objects to be protected, as taught by Barkley in his "isomorphic representation" -- may result in millions of security objects; Specification, p. 14, line 8; p. 18, lines 11 - 12. By contrast, Appellants' role-based organization may reduce the number of security objects in the same setting to several thousand security objects. Specification, p. 18, lines 9 - 11.)

30. Accordingly, as demonstrated by paragraphs 23 - 29 herein, Barkley fails to teach the second element of Appellants' independent Claim 19. Independent Claims 23 and 25 specify analogous limitations.

31. In view of paragraphs 18 - 30, Appellants respectfully submit that the Office Action fails to cite a reference that teaches each and every element of Appellants' independent Claims 19, 23, and 25, and fails to cite a reference that teaches all words of the claim language of these claims.

32. Appellants therefore respectfully submit that the Office Action fails to make out a *prima facie* case of anticipation as to independent Claims 19, 23, and 25, in violation of the above-quoted MPEP §706.02, §2131, and §2143.03. Without more, these claims are deemed patentable. See ***In re Oetiker***, 24 USPQ 2d 1443, 1444 (Fed. Cir. 1992), which stated:

> If the examination at the initial stage does not produce a prima facie case of

unpatentability, then without more the applicant is entitled to grant of the patent.

7.2) Rejection of Dependent Claims 3 - 12

33. Dependent Claims 3 - 12 stand or fall with independent Claim 19, from which they depend. Thus, these claims are deemed allowable by virtue of the allowability of the independent claim.

7.3) Rejection of Dependent Claims 20, 22, 24, and 26

34. Page 4, lines 1 - 10 of the Office Action discuss dependent Claims 20, 22, 24, and 26 and cite col. 8, lines 24 - 44 and col. 11, line 20 - col. 12, line 50. By incorporation of the independent claims from which they depend, each of these dependent claims pertains to "security objects [each of which] corresponds to a single role" (Claim 19, lines 3 - 4).

35. By contrast, the cited text from col. 8, lines 24 - 44 of Barkley discuss "an OAT assigned to that object" (col. 8, lines 35 - 36, emphasis added). An OAT that is assigned to an object is different from a security object that corresponds to a role (as discussed above in paragraph 29). Furthermore, the cited text in col. 11, line 20 - col. 12, line 50 pertains to OATs that are organized according to an object, where the OAT for a particular object may specify information for multiple roles, as has been discussed above in paragraphs 20 - 21. See, for example, col. 11, lines 40 - 45, stating that the "OAT accounts applies to files ... and to directories ... [and that] Various roles have varied permission with respect to these files [in the single "account" OAT]" (emphasis added). In other words, this single "account" OAT specifies permissions for more

than one role, in contrast to Appellants' claim language from Claim 19, lines 3 - 4. (In particular, this "account" OAT specifies permissions for 4 of the 5 roles illustrated in Table I, as discussed above in paragraph 20, fifth bullet.) Col. 11, lines 47 - 49 discusses permissions given to "all roles/groups" for the OAT "cd_to_dir"; clearly, a single OAT that specifies permissions for "all" roles (and in this particular example, the 5 roles illustrated in Table I) is distinct from Appellants' security objects that each pertain to a single role. The text in Col. 12, lines 46 - 49 also indicates that Barkley's OATs can each represent multiple roles, in contrast to Appellants' security objects, by stating "... the members of a given role may be assigned differing permissions ... by being assigned membership in differing OATs", emphasis added (and this text aligns with Barkley's Table I, where -- by way of example -- the "employee" role is shown as a member of the "cd_to_dir" OAT, the "employee_read" OAT, and the "suggestions" OAT).

36. Because the cited text fails to teach each and every element of Appellants' dependent Claims 20, 22, 24, and 26, and fails to teach all words of this claim language, Appellants respectfully submit that the Office Action fails to make out a *prima facie* case of anticipation as to these dependent claims, and without more, these claims are deemed patentable. (These dependent claims are also deemed patentable by virtue of the allowability of the independent claims from which they depend.)

7.4) Rejection of Dependent Claim 21

37. Dependent Claim 21 stands or falls with dependent Claim 20, from which it depends. Thus, this claim is deemed allowable by virtue of the allowability of Claim 20, the patentability

of which is discussed above in paragraphs 34 - 36.

8) CONCLUSION

For the reasons set out above, Appellants respectfully contend that each appealed claim is patentable, and respectfully request that the Examiner's Final Rejection of appealed Claims 3 - 12 and 19 - 26 should be reversed.

Respectfully submitted,

Marcia L. Doubet

Marcia L. Doubet,
Attorney for Appellants
Reg. No. 40,999

Customer Number for Correspondence: 43168
Phone: 407-343-7586
Fax: 407-343-7587

CLAIMS APPENDIX

CLAIMS AS CURRENTLY PRESENTED:

Claims 1 - 2 (canceled)

Claim 3: The method according to Claim 19, wherein at least one of the resources is an executable method.

Claim 4: The method according to Claim 19, wherein at least one of the resources is a column of a database table.

Claim 5: The method according to Claim 19, wherein at least one of the resources is a row of a database table.

Claim 6: The method according to Claim 19, wherein at least one of the resources is a file and each of the at least one actions identified for the at least one resource are file access operations that can be performed on the file.

Claim 7: The method according to Claim 19, wherein at least one of the resources is a function call to a function of an executable program.

Claim 8: The method according to Claim 19, wherein at least one of the resources is an Enterprise JavaBean ("EJB") and each of the at least one actions identified for the at least one resource are methods that can be performed on the EJB.

Claim 9: The method according to Claim 19, wherein at least one of the resources is a servlet and each of the at least one actions identified for the at least one resource are methods that can be performed by the servlet.

Claim 10: The method according to Claim 19, wherein at least one of the resources is a Uniform Resource Identifier ("URI") and each of the at least one actions identified for the at least one resource are methods which reference the URI.

Claim 11: The method according to Claim 19, wherein at least one of the resources is a JavaServer Page ("JSP") and each of the at least one actions identified for the at least one resource are methods referenced from the JSP.

Claim 12: The method according to Claim 19, wherein at least one of the resources is any resource that is expressible to the security system and each of the at least one actions identified for the at least one resource are selected from a set of actions that are permitted on that resource.

Claims 13 - 18 (canceled)

Claim 19: A computer-implemented method for enforcing role-permission security administration using security objects stored in a security repository, comprising steps of:

storing, in a security repository, a plurality of security objects, wherein each of the security objects corresponds to a single role;

specifying, in each of the security objects, all permissions granted to the corresponding role, wherein each of the specified permissions identifies at least one resource and, for each resource, at least one action that can be performed on the resource by subjects granted the corresponding role, wherein selected ones of the resources are identified in the specified permissions of more than one of the security objects and wherein the specified permissions for at least one of the security objects identifies a plurality of resources and for each of the plurality of resources, at least one of the actions; and

using the stored security objects to determine whether run-time requests for performing actions on the resources can be granted.

Claim 20: The method according to Claim 19, where the using step further comprises, for each of the run-time requests, the steps of:

determining, for the run-time request, a requester from which the request was received, and a particular action being requested on a particular resource;

determining one or more roles granted to the requester; and

until determining that the request can be granted or exhausting the determined roles, iteratively accessing the security object corresponding to each one of the determined roles and if the accessed security object identifies the requested action on the requested resource, then determining that the request can be granted.

Claim 21: The method according to Claim 20, wherein the step of determining one or more roles further comprises the steps of:

using an identification of the requester as a user identification to consult a mapping that specifies, for each of a plurality of subjects, one or more roles associated therewith, wherein each of the subjects is specified as at least one of (1) an identification of one or more users and (2) an identification of one or more user groups, thereby determining each role associated with the identification of the requester;

determining one or more user groups of which the requester is a member; and

using each of the determined user groups as a user group identification to consult the mapping, thereby determining each role associated with the determined user groups.

Claim 22: The method according to Claim 19, where the using step further comprises, for each of the run-time requests, the steps of:

determining, for the run-time request, a requester from which the request was received, and a particular action being requested on a particular resource; and

determining that the run-time request can be granted only if the requester has been granted at least one of the roles which is required, according to the stored security objects, to perform the requested action on the requested resource.

Claim 23: A system for enforcing role-permission security administration using security objects stored in a security repository, comprising:

a security repository for storing a plurality of security objects, wherein each of the security objects corresponds to a single role;

means for specifying, in each of the security objects, all permissions granted to the corresponding role, wherein each of the specified permissions identifies at least one resource and, for each resource, at least one action that can be performed on the resource by subjects granted the corresponding role, wherein selected ones of the resources are identified in the specified permissions of more than one of the security objects and wherein the specified permissions for at least one of the security objects identifies a plurality of resources and for each of the plurality of resources, at least one of the actions; and

means for using the stored security objects to determine whether run-time requests for performing actions on the resources can be granted.

Claim 24: The system according to Claim 23, where the means for using further comprises means for performing, for each of the run-time requests, steps of:

determining, for the run-time request, a requester from which the request was received, and a particular action being requested on a particular resource;

determining one or more roles granted to the requester; and

until determining that the request can be granted or exhausting the determined roles, iteratively accessing the security object corresponding to each one of the determined roles and if the accessed security object identifies the requested action on the requested resource, then determining that the request can be granted.

Claim 25: A computer program product for enforcing role-permission security administration using security objects stored in a security repository, the computer program product comprising computer-readable code embodied on one or more computer-usable media, the computer-readable code comprising instructions that when executed on a computer cause the computer to:

store, in a security repository, a plurality of security objects, wherein each of the security objects corresponds to a single role;

specify, in each of the security objects, all permissions granted to the corresponding role, wherein each of the specified permissions identifies at least one resource and, for each resource, at least one action that can be performed on the resource by subjects granted the corresponding role, wherein selected ones of the resources are identified in the specified permissions of more than one of the security objects·and wherein the specified permissions for at least one of the security objects identifies a plurality of resources and for each of the plurality of resources, at least one of the actions; and

use the stored security objects to determine whether run-time requests for performing actions on the resources can be granted.

Claim 26: The computer program product according to Claim 25, where the instructions that cause the computer to use the stored security objects further comprise instructions that cause the computer, for each of the run-time requests, to:

determine, for the run-time request, a requester from which the request was received, and a particular action being requested on a particular resource; and

determine that the run-time request can be granted only if the requester has been granted at least one of the roles which is required, according to the stored security objects, to perform the requested action on the requested resource.

EVIDENCE APPENDIX

Appellants, the Appellants' legal representative, and the assignee have no personal knowledge of evidence requiring separate identification herein as bearing on this Appeal.

RELATED PROCEEDINGS APPENDIX

No related proceedings are personally known to Appellants, the Appellants' legal representative, or the assignee.